Intelligent classification using adaptive fuzzy logic systems

Vassilis S. Kodogiannis[1]
Ilias Petrounias[2]
John N. Lygouras[3]

[1] Harrow School of Computer Science, University of Westminster
[2] Manchester Business School, University of Manchester
[3] Department of Electrical & Computer Engineering, Democritus University of Thrace, Greece

Intelligent Classification using Adaptive Fuzzy Logic Systems

Vassilis S. Kodogiannis, *Member IEEE*, Ilias Petrounias and John N. Lygouras, *Member IEEE*

***Abstract*—Fuzzy systems are currently finding practical applications, ranging from "soft" regulatory control in consumer products to accurate modeling of non-linear systems. A novel approach, based on adaptive fuzzy logic systems, has been discussed in this paper. Its performance is evaluated through a simulation study, using metered data collected from a roadside microphone-array sensor at the Valle d'Aosta highway in north-western Italy. The results indicate that the fuzzy classifier based on the proposed defuzzification method, namely area of balance (AOB), provide more accurate classifications compared to other classifiers.**

***Index Terms*—Neural Networks, Fuzzy systems, Vehicle Classification.**

I. Introduction

A Fuzzy Logic System (FLS) is a system utilizing fuzzy set theory and its operations to solve a given problem. The FLS can be classified into three broad types: pure fuzzy logic systems, Takagi and Sugeno's fuzzy system and fuzzy logic systems with fuzzifier and defuzzifier [1]. This later type of FLS is the most widely used FLS. In this study, a multi-input multi-output (MIMO) FLS will be used as a composed FLS. In this way fewer rules are required. Defuzzification converts this area to a real value. The centroid calculation approach is a logical answer to defuzzification because it uses all information available to compute the output. Centroid defuzzification can be put into equation form as

$$\bar{y} = \frac{[\int_S y\mu_B(y)dy]}{[\int_S \mu_B(y)dy]} \quad (1)$$

where S is the support of $\mu_B(y)$. One problem of the centroid defuzzifier is intensive computation. The centre average (CA) defuzzifier, on the other hand, is easy to compute and use. The CA defuzzifier has the formula

$$y_c = \frac{\sum_{m=1}^{M} \bar{y}^m \mu_{B^m}(\bar{y}^m)}{\sum_{m=1}^{M} \mu_{B^m}(\bar{y}^m)} \quad (2)$$

where $\bar{y}^m$ is the centre of gravity of the fuzzy set B^m and M is the number of rules. The problem with the CA defuzzifier is that it suffers from not using the entire shape of the consequent membership function. Regardless of whether the max-min or max-product inference is used, the result of the CA defuzzifier is still the same, regardless of whether the shape is narrow or wide.

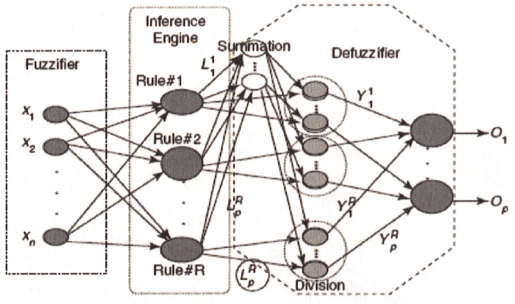

Fig. 1. AFLS concept

An adaptive fuzzy logic system (AFLS) is a fuzzy logic system having adaptive rules. Its structure is the same as a normal FLS but its rules are derived and extracted from given training data. In other words, its parameters can be trained like a neural network approach, but with its structure in a fuzzy logic system structure. The AFLS is one type of FLS with a singleton fuzzifier and centre average defuzzifier. The centroid defuzzifier cannot be used because of its computation expense and that it prohibits using the Backpropagation (BP) learning algorithm. The proposed in this research work AFLS consists of an alternative to classical defuzzification approaches, the area of balance (AOB) [2]. This AFLS has the same approach as the system presented by Wang [3] and its feed-forward structure is shown in Fig. 1 with an extra "fuzzy basis" layer. The fuzzy basis layer consists of fuzzy basis nodes for each rule. A fuzzy basis node has the following form:

$$\phi_m(\bar{x}) = \frac{\mu_m(\bar{x})}{\sum_{l=1}^{L} \mu_l(\bar{x})} \quad (3)$$

where $\varphi_m(\bar{x})$ is a fuzzy basis node for rule m and $\mu_m(\bar{x})$ is a membership value of rule m. Since we use a product-inference, the fuzzy basis node $\mu_m(\bar{x})$ is in the following form:

$$m(\bar{x}) = \prod_{i=1}^{n} \mu_{F_i^m}(x_i) \quad (4)$$

where $\mu_{F_i^m}(x_i)$ is a membership value of the i^{th} input of rule m. In case of a "Gaussian-shape" membership

Dr. V. S. Kodogiannis is in the Centre for Systems Analysis, School of Computer Science, University of Westminster, London HA1 3TP, UK (e-mail: kodogiv@wmin.ac.uk).

Dr. I. Petrounias is in the Manchester Business School, University of Manchester, Manchester, UK.

Assoc. Prof. J.N. Lygouras is in the Dept. of Electrical & Computer Engineering, Democritus University of Thrace, Greece

978-1-4244-1739-1/08/$25.00 © 2008 IEEE

function, then, $\mu_{F_i^m}(x_i)$ will be in the following form:

$$\mu_{F_{ii}^m}(x_i) = \exp\left[-\frac{(x_i - c_i^m)^2}{2(b_i^m)^2}\right] \quad (5)$$

where c_i^m and b_i^m are the centre and spread parameters, respectively, of the membership function i[th] input of the m[th] rule. The proposed in this paper defuzzification schemes will utilise Kosko's method with product inference [4]. The density (D) is defined as mass (M) per unit volume (V).

$$D = \frac{M}{V} \quad (6)$$

Let assume that we use the same material and all shapes have the same thickness, T, then

$$M = ATD \quad (7)$$

where A is an area and T is a thickness. Let us suppose that the shape of the membership function used in the consequent part is symmetric, e.g., triangular, Gaussian or bell-shape. The centre of gravity will pass through the halfway point of the base of that shape. For example, if we use a triangular shape and product-inference as a t-norm, then the shape of the consequent part of rule m will be shown as in Fig. 2. The shaded area (A) is

$$\frac{1}{2}\mu_m L_m .$$

Imagine that we have the consequent part of each rule placed on the mass-less beam having the pivot point at origin as shown in Fig. 3. Then,

$$F = M_1 g + M_2 g + M_3 g = (M_1 + M_2 + M_3)g \quad (8)$$

For balance

$$Fy = M_1 g y_b^1 + M_2 g y_b^2 + M_3 g y_b^3 = (M_1 y_b^1 + M_2 y_b^2 + M_3 y_b^3)g \quad (9)$$

$$y = \frac{(M_1 y_b^1 + M_2 y_b^2 + M_3 y_b^3)g}{F} = \frac{(M_1 y_b^1 + M_2 y_b^2 + M_3 y_b^3)g}{(M_1 + M_2 + M_3)g} \quad (10)$$

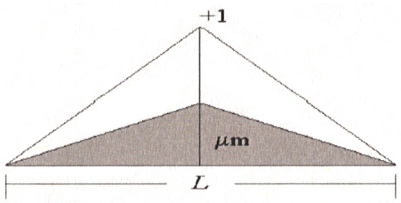

Fig. 2. Triangular shape membership function

Assume that D in Eq. 12 is the same, thus,

$$y = \frac{(A_1 y_b^1 + A_2 y_b^2 + A_3 y_b^3)TD}{(A_1 + A_2 + A_3)TD} = \frac{(A_1 y_b^1 + A_2 y_b^2 + A_3 y_b^3)}{(A_1 + A_2 + A_3)} \quad (11)$$

The area, A, will be depended upon the membership function used. Under the assumption of symmetric shape this method will have comparable capability with the centroid calculation method to approximate the output from the fuzzy set in the consequent part. If the triangle is used as a membership function and we use the max-product inference (Larsen logic), then the shaded area, A, will be derived as:

$$A_m = \frac{1}{2}\mu_m L_m \quad (12)$$

From Eq. 11, the output, y will be

$$y = \frac{(\mu_1 L_1 y_b^1 + \mu_2 L_2 y_b^2 + \mu_3 L_3 y_b^3)}{(\mu_1 L_1 + \mu_2 L_2 + \mu_3 L_3)} \quad (13)$$

In general form, the calculation of the output, y, will be

$$y_p = \frac{\sum_{m=1}^{M} \mu_m L_p^m y_p^m}{\sum_{m=1}^{M} \mu_m L_p^m} \quad (14)$$

where

y_p : the p[th] output of the network

μ_m: the membership value of the m[th] rule

L_p^m : the spread parameter of the membership function in the consequent part of the p[th] output of the m[th] rule

y_p^m : the centre of the membership function in the consequent part of the p[th] output of the m[th] rule.

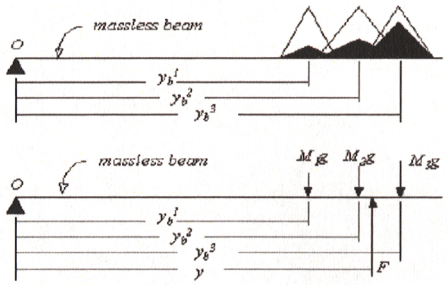

Fig. 3 Consequent fuzzy set placed on mass-less beam

A. *AFLS training with AOB Defuzzification*

Let us define the objective function as:

$$J = \sum_{k=1}^{K} J_k \quad (15)$$

where K is the number of training patterns, J_k is the sum of squared error for the k[th] pattern. Then, J_k is defined as:

$$J_k = \frac{1}{2}\sum_{p=1}^{P} (y_p(\bar{x}_k) - d_p(\bar{x}_k))^2 \quad (16)$$

where P is the number of outputs and d_p is the desired response of the p[th] output. $y_p(\bar{x}_k)$ is defined as in Eq. 14.

The update equation of y_p^m is as in the form:

$$y_p^m(n+1) = y_p^m(n) + m_y[y_p^m(n) - y_p^m(n-1)] - \eta_y \frac{\partial J}{\partial y_p^m}|_n \quad (17)$$

where

$$\frac{\partial J}{\partial y_p^m} = \frac{\partial J}{\partial J_m}\frac{\partial J_m}{\partial e_p}\frac{\partial e_p}{\partial y_p}\frac{\partial y_p}{\partial y_p^m} \quad 18)$$

$$= \sum_{k=1}^{K}\{(y_p^k - d_p^k)\frac{\mu_m^k L_p^m}{\sum_{m=1}^{M}\mu_m^k L_p^m} \quad 19)$$

where y_p^k is the p[th] output of the network corresponding to the k[th] pattern in the training data, d_p^k is the p[th] desired output of the k[th] pattern and μ_m^k is the membership value

of the m[th] rule corresponding to the k[th] pattern in the training data. The update equation of L_p^m is in the following form:

$$L_p^m(n+1)=L_p^m(n)+m_L[L_p^m(n)-L_p^m(n-1)]-\eta_L\frac{\partial J}{\partial L_p^m}|_n \quad (20)$$

where

$$\frac{\partial J}{\partial L_p^m}=\frac{\partial J}{\partial J_m}\frac{\partial J_m}{\partial e_p}\frac{\partial e_p}{\partial y_p}\frac{\partial y_p}{\partial L_p^m} \quad (21)$$

$$\frac{\partial J}{\partial L_p^m}=\sum_{k=1}^{K}\left[(y_p^k-d_p^k)\frac{\mu_m^k}{\left[\sum_{m=1}^{M}\mu_m^k L_p^m\right]}\left\{y_p^m-y_p^k\right\}\right] \quad (22)$$

where y_p^m is interpreted as a centre of the membership function of the p[th] output of the m[th] rule in the consequent part of IF-THEN rule. The update equation of the centre parameter, c_i^m, is in the form:

$$c_i^m(n+1)=c_i^m(n)+m_c[c_i^m(n)-c_i^m(n-1)]-\eta_c\frac{\partial J}{\partial c_i^m}|_n \quad (23)$$

where

$$\frac{\partial J}{\partial c_i^m}=\frac{\partial J}{\partial J_m}\frac{\partial J_m}{\partial e_p}\frac{\partial e_p}{\partial y_p}\frac{\partial y_p}{\partial \mu_m}\frac{\partial \mu_m}{\partial \mu_{F_i^m}}\frac{\partial \mu_{F_i^m}}{\partial c_i^m}+.... \quad (24)$$

In the present study, a "Gaussian-shape" function has been adopted, hence

$$\frac{\partial \mu_{F_i^m}}{\partial c_i^m}=\mu_{F_i^m}\frac{(x_i^k-c_i^m)}{b_i^{m^2}} \quad (25)$$

and

$$\frac{\partial \mu_m}{\partial \mu_{F_i^m}}=\prod_{\substack{j=1\\ j\neq i}}^{n}\mu_{F_i^j} \quad (26)$$

Thus,

$$\frac{\partial J}{\partial c_i^m}=\sum_{k=1}^{K}\left[\left\{\sum_{p=1}^{P}(y_p^k-d_p^k)\frac{L_p^m[y_p^m-y_p^k]}{\sum_{m=1}^{M}\mu_m^k L_p^m}\right\}\mu_m^k\frac{(x_i^k-c_i^m)}{b_i^{m^2}}\right] \quad (27)$$

The update equation of the spread parameter, b_i^m, is in the form:

$$b_i^m(n+1)=b_i^m(n)+m_b[b_i^m(n)-b_i^m(n-1)]-\eta_b\frac{\partial J}{\partial b_i^m}|_n \quad (28)$$

where

$$\frac{\partial J}{\partial b_i^m}=\frac{\partial J}{\partial J_m}\frac{\partial J_m}{\partial e_p}\frac{\partial e_p}{\partial y_p}\frac{\partial y_p}{\partial \mu_m}\frac{\partial \mu_m}{\partial \mu_{F_i^m}}\frac{\partial \mu_{F_i^m}}{\partial b_i^m}+.... \quad (29)$$

Again for the "Gaussian-shape" case,

$$\frac{\partial \mu_{F_i^m}}{\partial b_i^m}=\mu_{F_i^m}\frac{(x_i^k-c_i^m)^2}{b_i^{m^3}} \quad (30)$$

Therefore

$$\frac{\partial J}{\partial b_i^m}=\sum_{k=1}^{K}\left[\left\{\sum_{p=1}^{P}(y_p^k-d_p^k)\frac{L_p^m[y_p^m-y_p^k]}{\sum_{m=1}^{M}\mu_m^k L_p^m}\right\}\mu_m^k\frac{(x_i^k-c_i^m)^2}{b_i^{m^3}}\right] \quad (31)$$

All equations derived are used to update all parameters during the training phase of the network. The initial centre, c_i^m and y_{pm} are randomly selected from the k[th] training data, x_i^k and d_p^k respectively. The initial spread parameter, b_i^m, is determined by

$$b_i=\frac{\max(x_i)-\min(x_i)}{N} \quad (32)$$

Where b_i is a spread parameter of the i[th] input of all rules and N is the number of rules. Since the desired output in the classification problem is primarily a binary output representing each class, therefore, the initial spread parameter, L_p^m, can be set to 0.75. This method can be interpreted that we have initially equal confidence in each rule. This spread parameter will be adjusted during training.

II. Experimental Case

The overall acoustic signal of a vehicle arises from several sources, which include the engine, gears, fan, cooling system, road-tire interaction, exhaust, brakes, and aerodynamics effects [5]. However of great importance is not only the classification of acoustic signals, but also the source that generates the acoustic signals, such as vehicle classification and speaker verification. An acoustic classification system design composes of the acoustic sensor design and data acquisition, feature analysis and extraction, classifier design, and implementation. The vehicle acoustic classification system consists of several sub-systems as shown in Fig. 4.

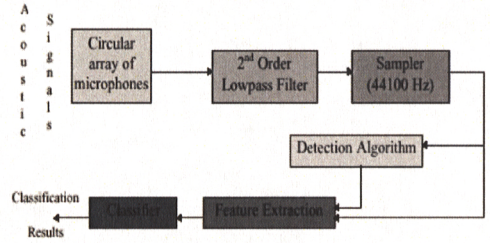

Fig. 4. Vehicle Acoustic Classification System

The definition of significant features plays an important role in the classification problem. In the vehicle classification problem, vehicles must be detected first before they can be classified. The detection process is designed to detect the presence of a passing vehicle and to initialize and end the feature extraction process. Since different classes of vehicles have different sizes, the ending point of feature extraction was adapted according to an initial prediction by a fuzzy logic system. The information of duration and loudness are used as the inputs to this prediction system. After a vehicle is detected significant features are extracted. All features are time domain features extracted directly from acoustic energy information. The vehicle acoustic signals have a wide range in frequencies as

shown in Fig. 5. Most of the acoustic energy is in low frequencies. Incoming signals are processed one window at a time.

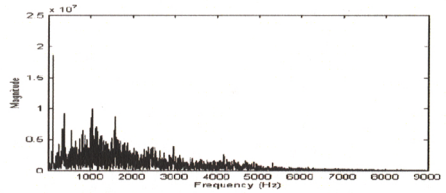

Fig. 5. Magnitude response of an acoustic signal

A rectangular window is used in this case. The size of the window is 0.02 seconds. Since the sampling rate is 44100 Hz, therefore, there are 882 samples in one window. The signal in this window is filtered by a high-pass filter having a cut-off frequency of 2700 Hz. Then, the filtered signals are filtered again by a low-pass filter having a cut-off frequency of 5400 Hz. Using these filters, the circular array ideally forms a beam pattern corresponding to the signals having frequencies from 2700 Hz to 5400 Hz. The energy (E) is calculated for each window. The energy (E) is defined by

$$E = \frac{\sum_{k=1}^{N} s^2(k)}{N} \tag{33}$$

where E is the energy in the window, $s(k)$ is the k^{th} sampled signal in that window, and N is the numbers of samples in the window. The current window is designed to overlap 50% with the previous window. The energy envelope of a typical passenger car is shown in Fig. 6. In Fig. 6(a), the original signals sampled with 44.1 kHz are shown. The filtered signals are shown in Fig. 6(c). In Fig. 6(b), the energy calculated in each window is shown.

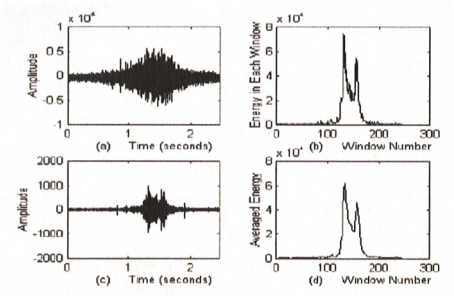

Fig. 6. Passenger car information (a) the original signals sampled with 44.1 kHz, (b) energy (E), (c) filtered signals, (d) average energy (avE)

The average energy *(avE)* of the energy E in the current window and five previous windows is shown in Fig. 6(d). The average energy *(avE)* is used in the detection task, while the energy E in each window is used mostly in the classification task. Besides the energy E information in the whole band of frequencies, the upper band of frequencies, 4500 to 5400 Hz, is selected. Fifth-order high-pass filter filters the filtered signals used in the detection. The average energies avE_2 of this band of frequencies are also calculated and used in the feature extraction process. Features will be extracted from both frequency bands, *e.g.*, from 2700 to 5400 Hz and from 4500 to 5400 Hz.

The detection algorithm was designed to detect not only one vehicle but also to initialize a feature extraction process for vehicle classification. Although different vehicle classes may have different sizes, the detection algorithm has to overcome this problem and reduce an under count and over count rate.

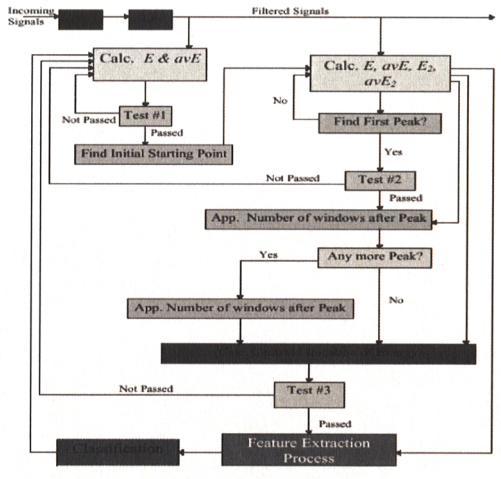

Fig. 7. Detection and feature extraction diagram

The under-count means that the algorithm does not detect the presence of a vehicle. It happens often to a small vehicle such as a passenger car. The over-count means that the algorithm can detect one vehicle as two vehicles. It happens often to a large vehicle such as a tractor-trailer truck. The over-count affects not only on the vehicle count information but also the classification performance. In the over-count situation two different sets of features are extracted for one vehicle. These two sets of features may not be the same, as one set of features and the classifier may not classify them correctly. In the under count situation the vehicle is definitely not classified.

In this study, the detection algorithm was designed to operate under normal highway traffic operations. The detection and feature extraction process is shown in Fig. 7. There are three tests in the designed detection algorithm. All tests are designed to overcome under and over count problems. For the first test three requirements should be met. First, the average acoustic energy *avE* must be above a pre-set threshold. This pre-set threshold is set such that most of average energy *avE* of small vehicles in adjacent lane is below this threshold. Second, the average acoustic energy *avE* must be rising. The final requirement is that the operation mode must be in reset mode. The reset mode is the mode that the sensor is ready to detect a new vehicle.

After the first test is passed, the feature extraction process begins. The initial starting extraction point is determined by using the average energy *avE* information. The initial starting point should be at the point the average energy *avE* is across the threshold. All necessary information used in feature extraction such as the energy avE_2 in the upper band must be initially gathered at this point. Then, the process gathers information when at least 0.1 seconds after the initial starting point have passed and when at least one of following conditions is met:

- the first condition is that the average energy *avE* is

below 125% of the previously pre-set threshold

- the second condition is that the average energy *avE* is decreasing
- the third condition is that the current window is more than 0.3 seconds away from the found peak

These conditions are set to make sure that the peak is found and there are few numbers of windows from the starting point. The indication of the energy movement indicating the energy rising or decreasing is calculated by the following equation:

$$Index = \frac{1}{N}\sum_{n=1}^{N}\frac{avE\ (k-N+n)}{avE\ (k-N+n-1)} \tag{34}$$

where avE represents the average energy, k the current window and N the number of window used, ($N = 8$ in this case). Once the first peak is found, the location of the peak, the peak value and the number windows above the pre-set threshold from the initial point are used in the second test. If all following conditions are met, then the second test is passed.

If any of them is not met, then the second test is failed and the operation is placed in the reset mode again. At this time, only features involving the energy envelope in time domain are considered. There are 30 features extracted as shown in Table 1.

TABLE 1: FEATURE DESCRIPTIONS

Feature number	Feature description
1	Maximum value of averaged energy (avE)
2	Approximated Location of centroid of energy from starting point
3	Location of the peak value of avE from the starting point
4	Difference in centroid location and the peak location
5	Maximum value of energy (E) in each window
6	Number of windows from starting point to ending point
7	Approximated Location of maximum value of E/avE in 1st region
8	Approximated Location of centroid of energy in 1st region
9	Mean of energy in 1st region
10	Approximated Location of maximum value of E/avE in 2nd region
11	Approximated Location of centroid of energy in 2nd region
12	Mean of energy in 2nd region
13	Approximated Location of maximum value of E/avE in 3rd region
14	Approximated Location of centroid of energy in 3rd region
15	Mean of energy in 3rd region
16	Number of windows having $E/avE > 1$
17	Number of windows having $avE > 50\%$ of maximum avE
18	Number of windows having $avE > 25\%$ of maximum avE
19	Maximum value of average energy, avE in 3rd region
20	Approximated Location of maximum value of E_2/avE_2 in 1st region
21	Approximated Location of maximum value of E_2/avE_2 in 2nd region
22	Approximated Location of maximum value of E_2/avE_2 in 3rd region
23	Number of windows having $E_2/avE_2 > 1$
24	Maximum value of avE_2
25	Sum of number of zeros crossing for each window in 1st region
26	Sum of number of zeros crossing for each window in 2nd region
27	Sum of number of zeros crossing for each window in 3rd region
28	Number of windows having $avE_2 > 50\%$ of maximum avE_2
29	Number of windows having $avE2 > 25\%$ of maximum avE_2
30	Number of windows after peak approximated by FLS

Fig. 8 shows an example of feature extraction along with the detection process. The "initial starting" shows the beginning of process after the first test is passed. The "centroid" is the centriod location of energy calculated as:

$$L = \frac{\sum_{k=1}^{N} kE\ (k)}{\sum_{k=1}^{N} E\ (k)} \tag{35}$$

where L represents the approximated location of the centroid of energy from starting point to the current position, N the number of windows from the starting point to the current one and $E(k)$ the energy of the k^{th} window. This location serves as the focal point of the detected vehicle's energies. The "starting" line is the beginning of the region of windows from which the features are extracted. The "ending" line is the end of the whole region. Besides the windows, there are three small regions: 1st, 2nd and 3rd region to extract features. The 1st region is defined as the region from the starting line to the centroid line. The 2nd region is defined as the region from the centroid line to halfway between the centroid and the ending line.

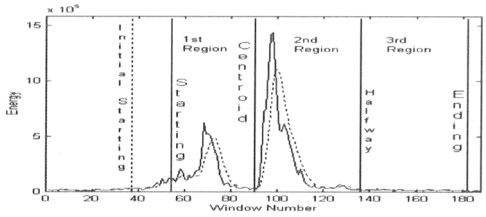

Fig. 8. Feature extraction regions.

The 3rd region is defined as the region from the halfway line to the ending line. All features are derived heuristically. Features number 7, 8, 9, 10, 11, 12, 13, 14, 15, 19, 20, 21, 22, 25, 26 and 27 are features extracted from each region. The others are from the whole region. The ratio between the energy $\boldsymbol{E}$ in each window and the averaged energy $\boldsymbol{avE}$ will indicate the fluctuation of energy. The approximate location of these fluctuations in each region indicates where the most fluctuation in that region is. The maximum value of energy indicates how loud it is. Most trucks are loud, but some passenger cars are also loud. The features involving the amplitude of energy, *e.g.,* feature number 5, are transformed into log scale. Then, all features are linearly transformed into a range from 0 to 2.

III. CLASSIFICATION WITH AFLS-AOB

Training and testing data used in this research are collected from the roadside sensor station and processed using the described detection and feature extraction. Metered data were collected from a roadside microphone-array sensor at the Valle d'Aosta highway in north-western Italy. The whole dataset consists of five classes. Class 1 includes passenger cars. Class 2 includes pickups, vans and mini-vans. Class 3 includes all single unit trucks having two axles, six tires. They are small trucks, small trucks with flatbeds, small trucks with flatbeds including loads, trash trucks, heavy-duty trucks, trucks with box, *etc*. Class 4 includes all single unit trucks having three axles, ten tires. They are heavy-duty trucks, trash trucks, cement trucks, trucks with box, *etc.* The last class, class 5, includes all five-axle single trailer trucks that include tractor-trailer trucks, gas trucks, flatbeds, and flatbeds with load. Class 1 has the most dominant number in the whole data and class 4, the fewest. Because of the weather during the data-collecting period, there are few numbers in class 4. These data were collected in various conditions, from cold to warm weather, from cloudy to clear sky and from early morning to late afternoon and represent 2440 passenger cars, 1007 pickups or vans, 587 two-axle six-tire trucks, 309 three-axle single unit trucks, and 963 five-axle single trailer trucks. The road surfaces are from dry to light wet surface. Data during snow and raining are excluded. All data are collected under normal highway traffic operations.

Vehicle acoustic data were collected, processed, and extracted from raw acoustic data as explained in the previous sections. For each vehicle, 30 features have been extracted. The five-class problem was the hardest case in this study. Each class consists of a variety of vehicles. For example, the two-axle six-tire class includes all small trucks having two axles and six tires. This class overlaps

with other classes, *e.g.,* pickups, heavy-duty trucks.

The standard MLP with the back-propagation learning algorithm widely used for such an application was considered in this work as a testbed case. This algorithm is based on the gradient descent concept. The numbers of MLP hidden nodes are determined by an increment method procedure. The network starts with a small number of hidden nodes then it is incremented until the outputs of the network saturated. The final number of hidden nodes is determined by the smallest network giving relatively the same results as a bigger one. The optimum structure has been determined to be with 2 hidden layers and with 20 and 12 nodes in the two layers [6].

However, the introduction of hybrid learning algorithms imposed a new dimension to this specific problem. The main advantage of the proposed AFLS method is the ability to learn from experience and a high computation rate. The average percentage relative error approaches its optimal value after short time training. This is due to the fact that the consequence parameters have converged. This implies that the convergence of consequence parameters play a dominant role in system estimation accuracy. The remaining time is just for fine-tuning the premise parameters. Thus the training required to achieve acceptable accuracy was very fast compared to the MLP method.

A. *Five-Class Problem*

The five-class problem is the toughest problem in this study. Each class consists of a variety of vehicles. For example, the two-axle six-tire class includes all small trucks having two axles and six tires. This class overlaps with other classes, *e.g.,* pickups, heavy-duty trucks. The classifier learned well in certain classes, and it would definitely classify well in those classes. The numbers of MLP hidden nodes are determined by an increment method procedure. The network starts with a small number of hidden nodes then it is incremented until the outputs of the network saturated. The final number of hidden nodes is determined by the smallest network giving relatively the same results as a bigger one. With the AFLS the numbers of rules are determined by a similar increment method procedure. The initial parameters of AFLS are initialized by the training data picked randomly. The input feature vector is used as the initial centers of the antecedent part. The centers of the consequent part are initialized by the corresponding desired output. In classification problems the desired outputs are normally in binary form. The Gaussian shape function is used as a membership function in AFLS. Results are shown in Table 2.

TABLE 2

	% Cl. 1	% Cl. 2	% Cl. 3	% Cl. 4	% Cl. 5	*Total correct*
MLP	74.26	65.87	70.07	71.43	89.63	***74.83***
AOB	83.11	59.13	75.51	85.71	93.78	***79.80***

B. *Four-Class Problem*

In a four-class problem, Classes 1 and 2 are combined as one small vehicle class. The remaining classes are the same as in previous problem. The number of rules is determined with the same procedures as in the previous problem. The number of rules is 14. In this problem the numbers of data are 3447, 587, 309, and 963 for Class #1, #2, #3, #4, respectively. Class 1 "dominates" the other classes. Again, the AFLS-AOB has better results than the other classifiers as illustrated in Table 3.

TABLE 3

	% Class 1	% Class 2	% Class 3	% Class 4	*Total correct*
MLP	95.82	59.86	76.62	88.38	***89.37***
AOB	95.59	79.59	79.22	92.12	***92.24***

C. *Small versus Large Vehicle Classification*

In this problem vehicles in Classes 1 and 2 are combined as a small vehicle class and Classes 3, 4, and 5 are combined as a large vehicle class. There will be 3447 in one class and 1859 in the other class. Again the AFLS-AOB has better classification performances than the MLP. Table 4 shows the related results.

TABLE 4

	% Cl. 1	% Cl. 2	*Total % correct*
MLP	95.33	95.48	***96.68***
AOB	97.80	97.20	***97.59***

IV. CONCLUSIONS

This paper involves design of classification system design for vehicles detection using their acoustic signature. This study was focused mainly on features analysis and extraction, and the classifier system design. A detection algorithm has been developed to detect a presence of a passing vehicle. This detection algorithm initialized and ended the feature extraction process. After a vehicle was detected significant features were extracted. The advantage of using an AFLS over a MLP is that its parameters can be initialized more effectively than MLPs. With good initialization the AFLS trains much faster than a corresponding MLP. It demonstrated possibility of an acoustic sensor system with 97.59% correct classification rate between small and large vehicles on 1327 vehicles, 92.24% correct classification rate in four-class problem on 1327 vehicles, and 79.80% correct classification rate in five-class problem on 1327 vehicles.

REFERENCES

[1] T Ross, "*Fuzzy Logic with Engineering Applications*", McGraw Hill, 1995.

[2] V.S. Kodogiannis, M. Boulougoura, E. Wadge, J.N. Lygouras, "The usage of soft-computing methodologies in interpreting capsule endoscopy", *Engineering Applications in Artificial Intelligence*, vol. 20, pp. 539-553, 2007

[3] Wang, L.X., "*Adaptive Fuzzy Systems and Control*", Prentice Hall, Inc., 1994.

[4] Kosko, B., "*Neural Networks and Fuzzy Systems: A Dynamical Systems Approach to Machine Intelligence*", Prentice Hall, 1992

[5] C. Harlow and S. Peng, Automatic vehicle classification system with range sensors, *Transportation Research C9*, pp. 231–247, 2001.

[6] V.S. Kodogiannis, "Comparison of advanced learning algorithms for short-term load forecasting," *Journal of Intelligent & Fuzzy Systems*, vol. 8, No. 4, pp. 243-259, 2000.